Amendments to the Specification:

Replace the paragraph beginning at page 6, line 14 with the following amended paragraph:

-- One example of an application for the hardware-based multithreaded processor 12 is as a network processor. As a network processor, the hardware-based multithreaded processor 12 interfaces to network devices such as a Media Access Controller (MAC) device, e.g., a 10/100BaseT Octal MAC ~~13a~~ or a Gigabit Ethernet device ~~13b~~. In general, as a network processor, the hardware-based multithreaded processor 12 can interface to any type of communication device or interface that receives or sends large amount of date. The computer processing system 10 functioning in a networking application can receive network packets and process those packets in a parallel manner. --

Replace the paragraph beginning at page 9, line 3 as with the following amended paragraph:

-- The GPRs 32 are used for general programming purposes. The GPRs 32 are read and written exclusively under program control. The GPRs 32, when used as a source in an instruction, supply operands to an execution datapath 44. When used as a destination in an instruction, the GPRs 32 are written with the result of the execution datapath 44. The programming engine 16a also includes I/O transfer registers 34, 36, 38, and 40 which are used for transferring data to and from the programming engine 16a and locations external to the programming engines 16a, e.g., the DRAM memory 14a, the SRAM memory 14b, ~~and~~ etc. --

Replace the paragraph beginning at page 18, line 22 as with the following amended paragraph:

-- On the other hand, if the CAM 64 indicates a miss during the read phase 122, a read of the needed variable is initiated (130). Consequently, the execution time of the remaining seven (7) contexts is being used to completely hide the latency of the read (132). Moreover, the variable is available at the modify write stage of this context (132). The write latency of the

critical section is avoided since the variable is already valid in the CAM 64 if recently used. Next, the CAM is written or updated (134). The content of the CAM 64 provides the location of the LRU cached variable, with the new variable overriding the previously used variable (136). --